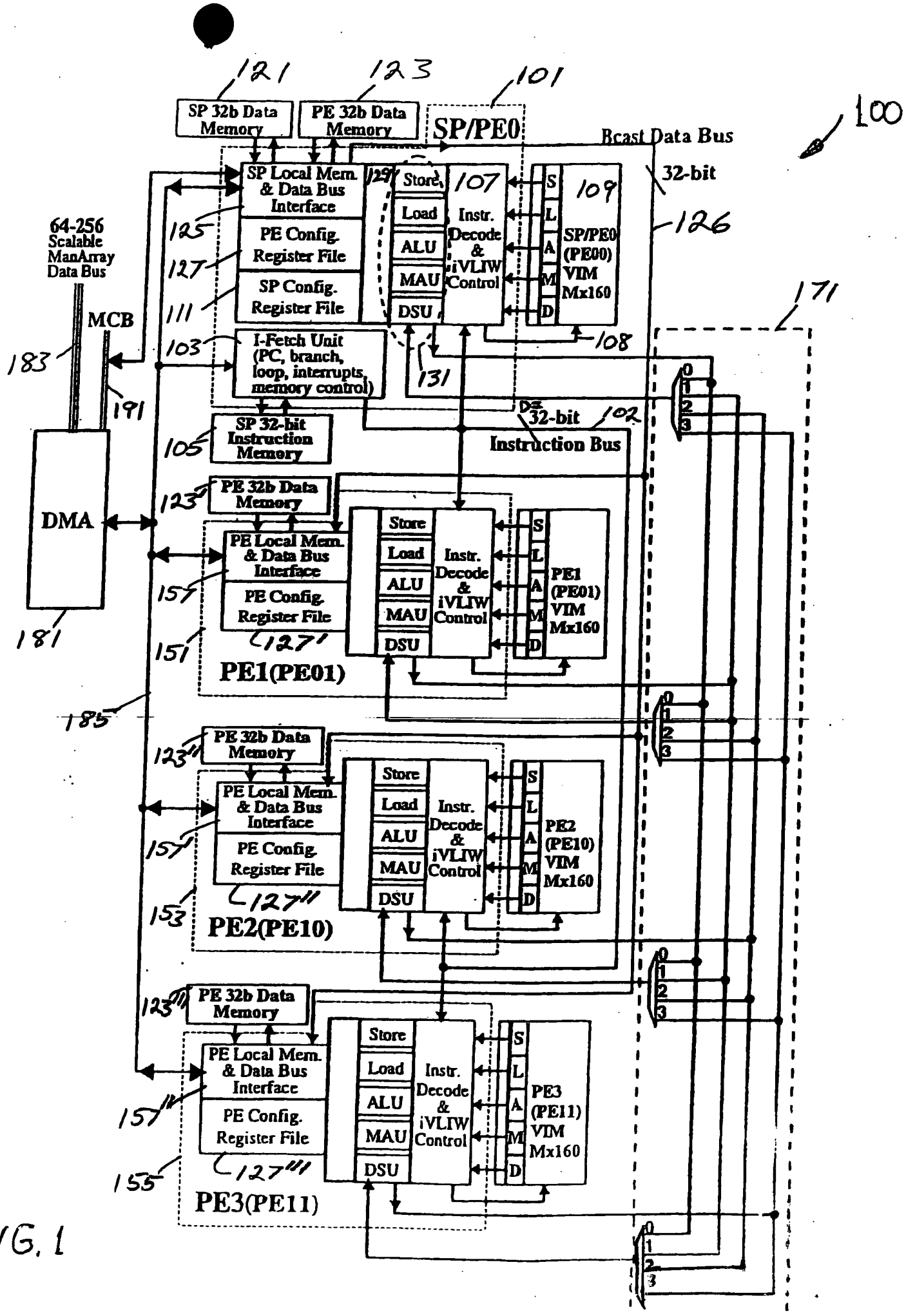
100
121
123
101
SP 32b Data Memory
PE 32b Data Memory
SP/PE0
Bcast Data Bus
32-bit
126
SP Local Mem. & Data Bus Interface
PE Config. Register File
SP Config. Register File
Store
Load
ALU
MAU
DSU
Instr. Decode & iVLIW Control
107
109
S
L
A
M
D
SP/PE0 (PE00) VIM Mx160
125
127
111
129
64-256 Scalable ManArray Data Bus
MCB
183
191
I-Fetch Unit (PC, branch, loop, interrupts, memory control)
103
131
108
171
SP 32-bit Instruction Memory
105
32-bit
102
Instruction Bus
DMA
181
PE 32b Data Memory
123
PE Local Mem. & Data Bus Interface
PE Config. Register File
157
151
127'
PE1 (PE01) VIM Mx160
PE1(PE01)
185
123''
153
127''
PE2 (PE10) VIM Mx160
PE2(PE10)
123'''
155
127'''
PE3 (PE11) VIM Mx160
PE3(PE11)
0
1
2
3

FIG. 1

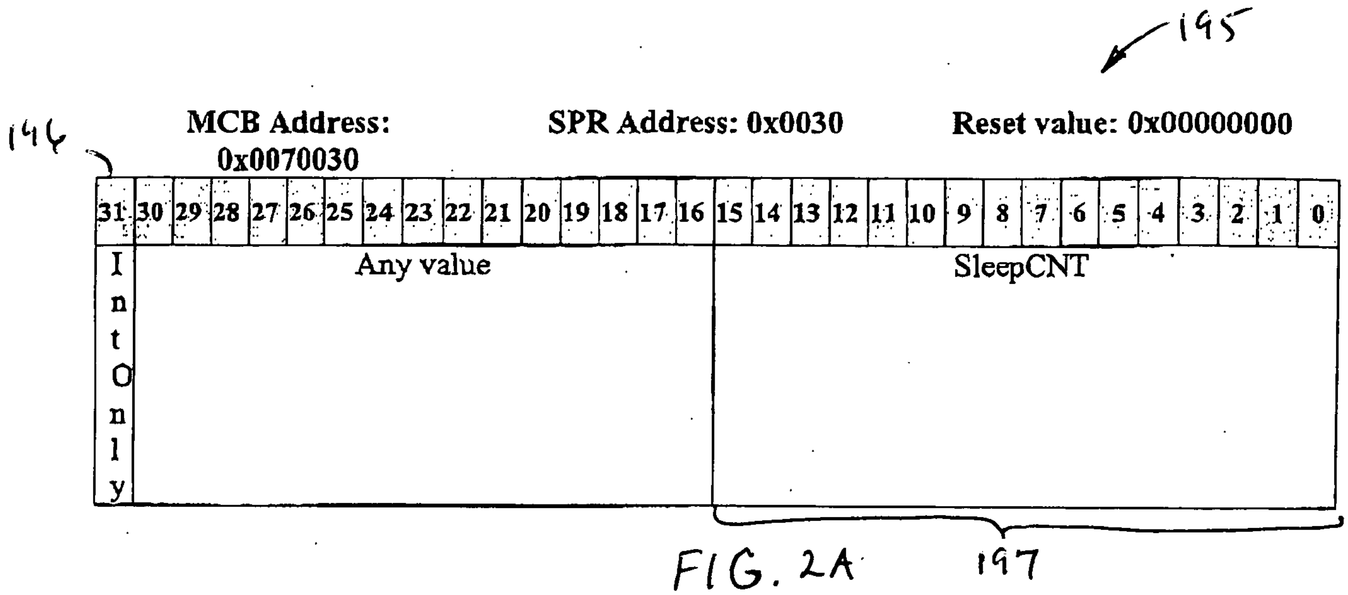
195
146
MCB Address: 0x0070030
SPR Address: 0x0030
Reset value: 0x00000000
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
IntOnly
Any value
SleepCNT
197

FIG. 2A

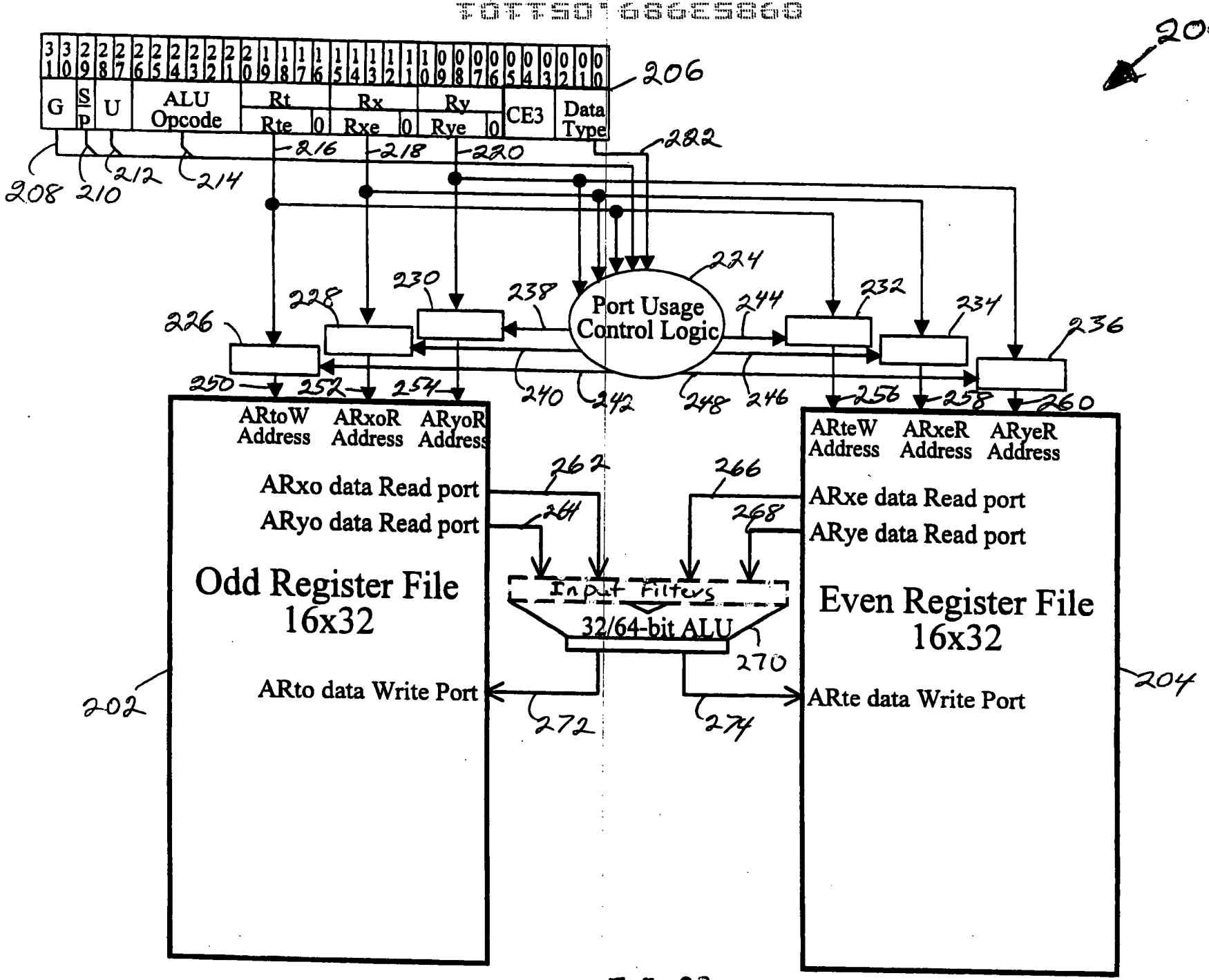
200
206
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
G
SP
U
ALU Opcode
Rt
Rte
0
Rx
Rxe
0
Ry
Rye
0
CE3
Data Type
208
210
212
214
216
218
220
222
224
226
228
230
238
Port Usage Control Logic
244
232
234
236
250
252
254
240
242
248
246
256
258
260
ARtoW Address
ARxoR Address
ARyoR Address
ARxo data Read port
ARyo data Read port
Odd Register File 16x32
ARto data Write Port
202
262
264
266
268
Input Filters
32/64-bit ALU
270
272
274
ARteW Address
ARxeR Address
ARyeR Address
ARxe data Read port
ARye data Read port
Even Register File 16x32
ARte data Write Port
204

FIG. 2B

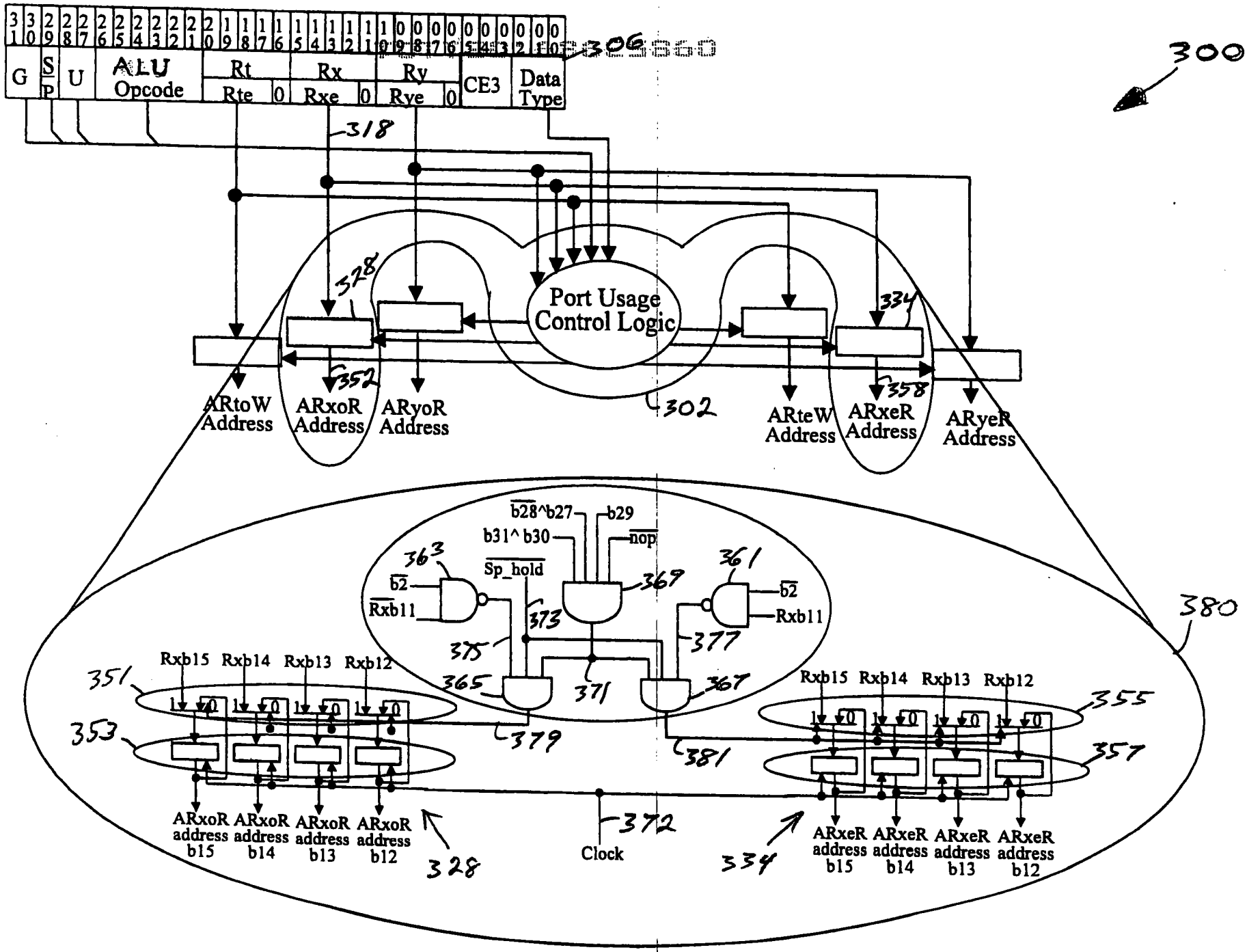
300
306
G
S/P
U
ALU Opcode
Rt
Rte
0
Rx
Rxe
0
Ry
Rye
0
CE3
Data Type
318
328
Port Usage Control Logic
302
334
352
358
ARtoW Address
ARxoR Address
ARyoR Address
ARteW Address
ARxeR Address
ARyeR Address
380
b28^b27
b29
b31^ b30
nop
Sp_hold
363
361
369
373
375
377
365
371
367
379
381
b2
Rxb11
Rxb15 Rxb14 Rxb13 Rxb12
351
353
355
357
ARxoR address b15
ARxoR address b14
ARxoR address b13
ARxoR address b12
Clock
372
334
ARxeR address b15
ARxeR address b14
ARxeR address b13
ARxeR address b12

FIG. 3A

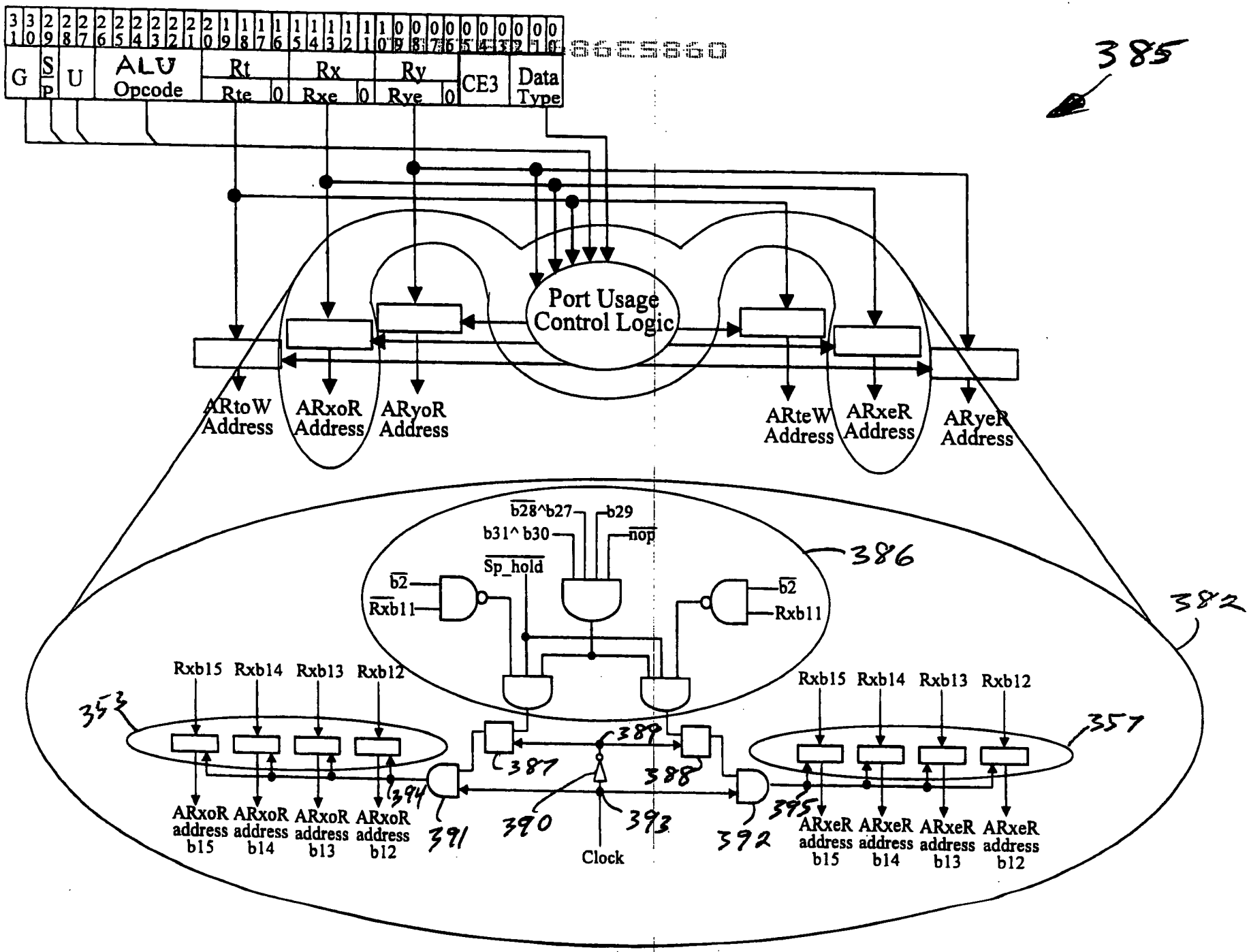
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
G
SP
U
ALU Opcode
Rt
Rte 0
Rx
Rxe 0
Ry
Rye 0
CE3
Data Type
385
Port Usage Control Logic
ARtoW Address
ARxoR Address
ARyoR Address
ARteW Address
ARxeR Address
ARyeR Address
386
382
b28^b27
b29
b31^ b30
nop
Sp_hold
b2
Rxb11
Rxb15 Rxb14 Rxb13 Rxb12
353
357
387
388
389
390
391
392
393
394
395
Clock
ARxoR address b15
ARxoR address b14
ARxoR address b13
ARxoR address b12
ARxeR address b15
ARxeR address b14
ARxeR address b13
ARxeR address b12

FIG. 3B

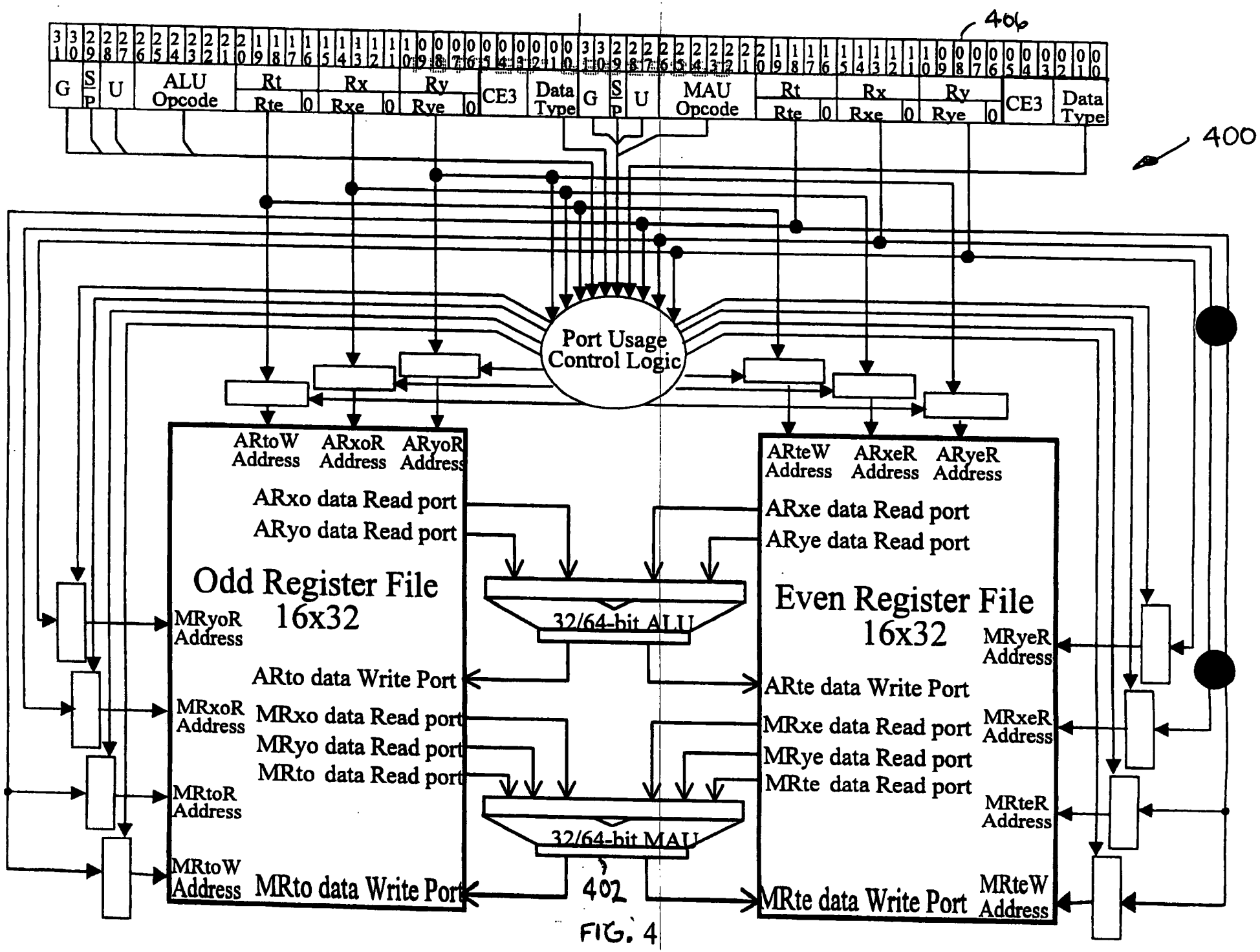
406
400
G
SP
U
ALU Opcode
Rt
Rte
0
Rx
Rxe
0
Ry
Rye
0
CE3
Data Type
G
SP
U
MAU Opcode
Rt
Rte
0
Rx
Rxe
0
Ry
Rye
0
CE3
Data Type
Port Usage Control Logic
ARtoW Address
ARxoR Address
ARyoR Address
ARxo data Read port
ARyo data Read port
Odd Register File 16x32
MRyoR Address
ARto data Write Port
MRxoR Address
MRxo data Read port
MRyo data Read port
MRto data Read port
MRtoR Address
MRtoW Address
MRto data Write Port
32/64-bit ALU
32/64-bit MAU
402
ARteW Address
ARxeR Address
ARyeR Address
ARxe data Read port
ARye data Read port
Even Register File 16x32
MRyeR Address
ARte data Write Port
MRxeR Address
MRxe data Read port
MRye data Read port
MRte data Read port
MRteR Address
MRte data Write Port
MRteW Address

FIG. 4

ALU Operation Encoding

500

31–30	29	28–27	26–21	20–17	16	15–12	11	10–7	6	5	4–3	2–0
Group	S/P	Unit	ALUopcode	Rt		Rx		Ry		0	CE2	DPack
			MAUopcode	Rte	0	Rxe	0	Rye	0			

503 501

Arithmetic Scalar Flags Affected (on least significant operation)
C = 1 if a carry occurs, 0 otherwise
N = MSB of result
V = 1 if an overflow occurs, 0 otherwise
Z = 1 if result is zero, 0 otherwise
See also ASF Definitions in chapter on Conditional Execution.
Cycles: 1

Arithmetic Execution Unit – 504
00 = ALU
01 = MAU
10 = DSU
11 = Reserved
$b_{28} b_{27}$

DPack – 502

Integer Data Packing
000 = 4 Bytes (4B)
001 = 2 Halfwords (2H)
010 = 1 Word (1W)
011 = Reserved
100 = 8 Bytes (8B)
101 = 4 Halfwords (4H)
110 = 2 Words (2W)
111 = 1 Doubleword (1D)
$b_2 b_1 b_0$

FIG. 5A

Description

The sum of source registers *Rx* and *Ry* is stored in target register *Rt*.

Syntax/Operation

520

Instruction	Operands	Operation	ACF
			Doubleword
ADD.[SP][AM].1D	Rte, Rxe, Rye	Rto||Rte ← Rxo||Rxe + Ryo||Rye	None
[TF].ADD.[SP][AM].1D	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None
			Word
ADD.[SP][AM].1W	Rt, Rx, Ry	Rt ← Rx + Ry	None
[TF].ADD.[SP][AM].1W	Rt, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None
			Dual Words
ADD.[SP][AM].2W	Rte, Rxe, Rye	Rto ← Rxo + Ryo Rte ← Rxe + Rye	None
[TF].ADD.[SP][AM].2W	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None
			Dual Halfwords
ADD.[SP][AM].2H	Rt, Rx, Ry	Rt.H1 ← Rx.H1 + Ry.H1 Rt.H0 ← Rx.H0 + Ry.H0	None
[TF].ADD.[SP][AM].2H	Rt, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None
			Quad Halfwords
ADD.[SP][AM].4H	Rte, Rxe, Rye	Rto.H1 ← Rxo.H1 + Ryo.H1 Rto.H0 ← Rxo.H0 + Ryo.H0 Rte.H1 ← Rxe.H1 + Rye.H1 Rte.H0 ← Rxe.H0 + Rye.H0	None
[TF].ADD.[SP][AM].4H	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None
			Quad Bytes
ADD.[SP][AM].4B	Rt, Rx, Ry	Rt.B3 ← Rx.B3 + Ry.B3 Rt.B2 ← Rx.B2 + Ry.B2 Rt.B1 ← Rx.B1 + Ry.B1 Rt.B0 ← Rx.B0 + Ry.B0	None
[TF].ADD.[SP][AM].4B	Rt, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None
			Octal Bytes
ADD.[SP][AM].8B	Rte, Rxe, Rye	Rto.B3 ← Rxo.B3 + Ryo.B3 Rto.B2 ← Rxo.B2 + Ryo.B2 Rto.B1 ← Rxo.B1 + Ryo.B1 Rto.B0 ← Rxo.B0 + Ryo.B0 Rte.B3 ← Rxe.B3 + Rye.B3 Rte.B2 ← Rxe.B2 + Rye.B2 Rte.B1 ← Rxe.B1 + Rye.B1 Rte.B0 ← Rxe.B0 + Rye.B0	None
[TF].ADD.[SP][AM].8B	Rte, Rxe, Rye	Do operation only if T/F condition is satisfied in F0	None

FIG. 5B

MPYA - Multiply Accumulate

Encoding

660

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Group		S/P	Unit		MAUopcode						Rte				0		Rx				Ry					CE3	MPack					

FIG. 6A

Syntax/Operation

620

Instruction	Operands	Operation	ACF
			Word
MPYA.[SP]M.1[SU]W	Rte, Rx, Ry	Do operation below but do not affect ACFs	None
MPYA[CNVZ].[SP]M.1[SU]W	Rte, Rx, Ry	Rto||Rte ← Rto||Rte + (Rx * Ry)	F0
[TF].MPYA.[SP]M.1[SU]W	Rte, Rx, Ry	Do operation only if T/F condition is satisfied in ACFs	None
			Dual Halfwords
MPYA.[SP]M.2[SU]H	Rte, Rx, Ry	Do operation below but do not affect ACFs	None
MPYA[CNVZ].[SP]M.2[SU]H	Rte, Rx, Ry	Rto ← Rto + (Rx.H1 * Ry.H1) Rte ← Rte + (Rx.H0 * Ry.H0)	F1 F0
[TF].MPYA.[SP]M.2[SU]H	Rte, Rx, Ry	Do operation only if T/F condition is satisfied in ACFs	None
			Quad Bytes
MPYA.[SP]M.4[SU]B	Rte, Rx, Ry	Do operation below but do not affect ACFs	None
MPYA[CNVZ].[SP]M.4[SU]B	Rte, Rx, Ry	Rto.H1 ← Rto.H1 + (Rx.B3 * Ry.B3) Rto.H0 ← Rto.H0 + (Rx.B2 * Ry.B2) Rte.H1 ← Rte.H1 + (Rx.B1 * Ry.B1) Rte.H0 ← Rte.H0 + (Rx.B0 * Ry.B0)	F3 F2 F1 F0
[TF].MPYA.[SP]M.4[SU]B	Rte, Rx, Ry	Do operation only if T/F condition is satisfied in F0	None

FIG. 6B

640

Arithm tic Scalar Flags Affected (on least significant operation)

C = Not affected
N = MSB of result
V = Not affected
Z = 1 if result is zero, 0 otherwise
Cycles: 2

Arithm tic Execution Unit

00 = ALU
01 = MAU
10 = DSU
11 = Reserved

$b_{28} b_{27}$

Instruction Group

00 = Reserved
01 = Flow Control
10 = Load/Store (LU, SU)
11 = Arithmetic/Logical (ALU, MAU, DSU)

$b_{31} b_{30}$

Mpack - Multiply Data Packing

000 = Reserved
001 = 2 Halfwords (2H)
010 = 1 Word (1W)
011 = Reserved
100 = Reserved
101 = 4 Halfwords (4H) for MPYH and MPYL
110 = Reserved
111 = Reserved

$b_2 b_1 b_0$

SP/PE Select

0 = SP
1 = PE

FIG. 6C

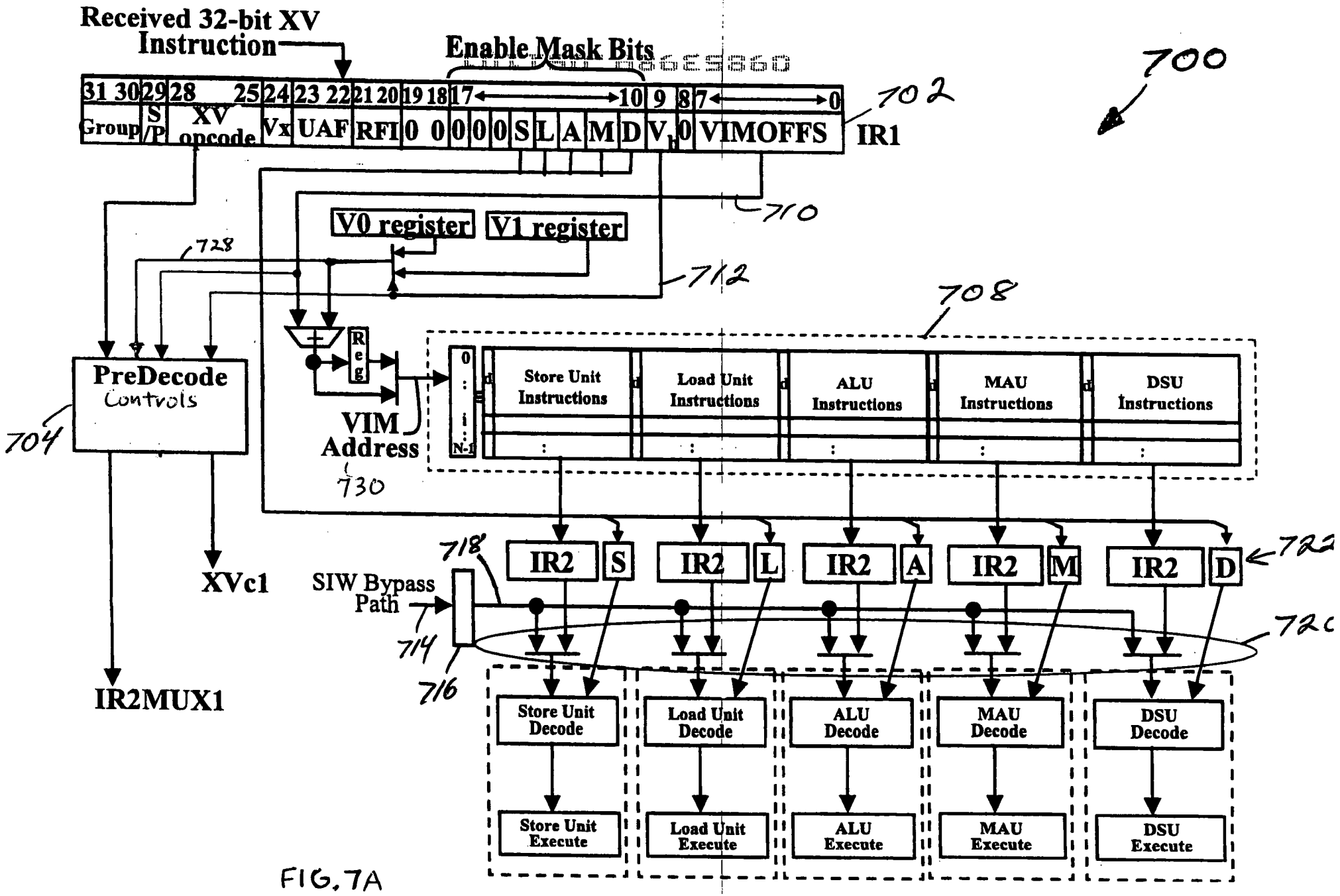
Received 32-bit XV Instruction
Enable Mask Bits
700
31 30 29 28 25 24 23 22 21 20 19 18 17 10 9 8 7 0
Group
S/P
XV opcode
Vx
UAF
RFI
0 0 0 0 0 S L A M D V_h 0
VIMOFFS
IR1
702
710
V0 register
V1 register
728
712
708
PreDecode Controls
704
Reg
VIM Address
730
0
N-1
Store Unit Instructions
Load Unit Instructions
ALU Instructions
MAU Instructions
DSU Instructions
XVc1
718
IR2
S
IR2
L
IR2
A
IR2
M
IR2
D
SIW Bypass Path
714
716
IR2MUX1
Store Unit Decode
Load Unit Decode
ALU Decode
MAU Decode
DSU Decode
Store Unit Execute
Load Unit Execute
ALU Execute
MAU Execute
DSU Execute

FIG. 7A

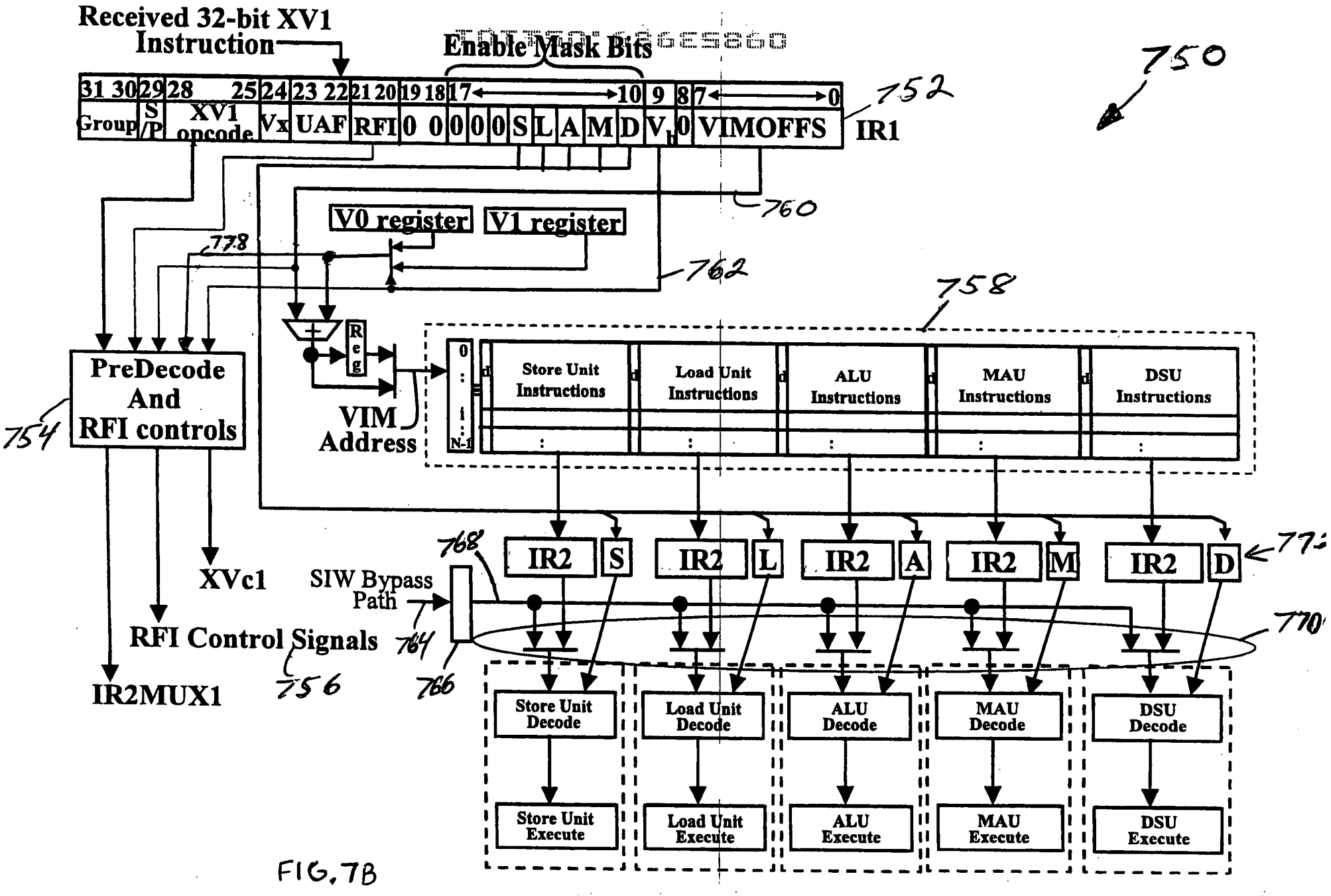
Received 32-bit XV1 Instruction
Enable Mask Bits
750
31 30 29 28 25 24 23 22 21 20 19 18 17 10 9 8 7 0
Group
S/P
XV1 opcode
Vx
UAF
RFI
0 0 0 0 0 S L A M D Vh 0
VIMOFFS
752
IR1
760
V0 register
V1 register
778
762
758
PreDecode And RFI controls
754
Reg
VIM Address
0
N-1
Store Unit Instructions
Load Unit Instructions
ALU Instructions
MAU Instructions
DSU Instructions
IR2
S
L
A
M
D
772
768
XVc1
SIW Bypass Path
RFI Control Signals
IR2MUX1
764
756
766
770
Store Unit Decode
Load Unit Decode
ALU Decode
MAU Decode
DSU Decode
Store Unit Execute
Load Unit Execute
ALU Execute
MAU Execute
DSU Execute

FIG. 7B

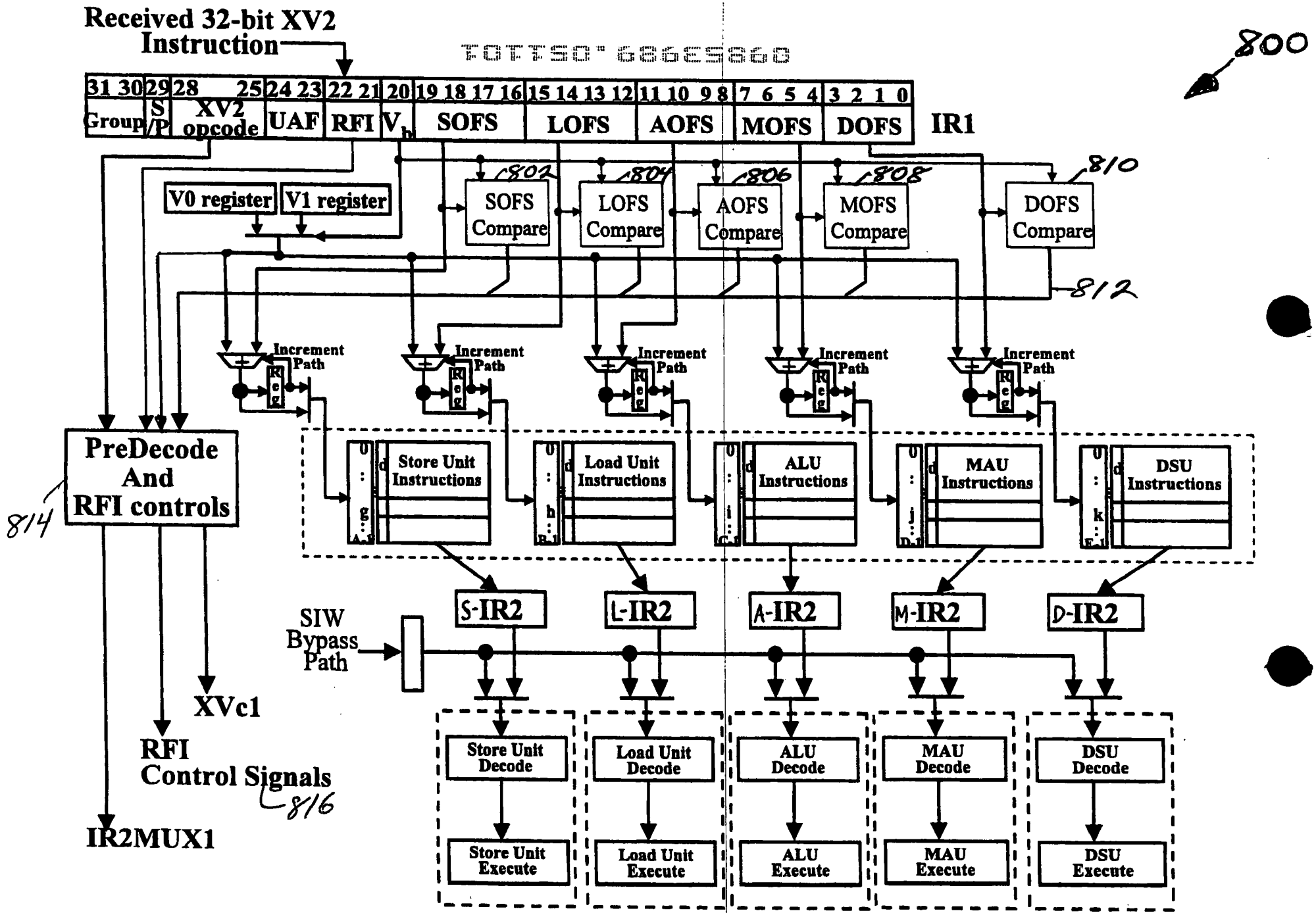
Received 32-bit XV2 Instruction
800
31 30 29 28 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group
S/P
XV2 opcode
UAF
RFI
V_h
SOFS
LOFS
AOFS
MOFS
DOFS
IR1
V0 register
V1 register
802
804
806
808
810
SOFS Compare
LOFS Compare
AOFS Compare
MOFS Compare
DOFS Compare
812
Increment Path
Reg
PreDecode And RFI controls
814
Store Unit Instructions
Load Unit Instructions
ALU Instructions
MAU Instructions
DSU Instructions
S-IR2
L-IR2
A-IR2
M-IR2
D-IR2
SIW Bypass Path
XVc1
RFI Control Signals
816
IR2MUX1
Store Unit Decode
Load Unit Decode
ALU Decode
MAU Decode
DSU Decode
Store Unit Execute
Load Unit Execute
ALU Execute
MAU Execute
DSU Execute

FIG. 8